УДК 004.05:004.416.3:004.422.63

В.И. Шинкаренко, Г.В. Забула
Днепропетровский национальный университет железнодорожного транспорта
МОН Украины, г. Днепропетровск
Украина, 49010, г. Днепропетровск, ул. акад. Лазаряна, 2

Применение генетического алгоритма в задачах адаптации структур данных

V.I. Shynkarenko, H.V.Zabula
Dnipropetrovsk National University of Railway Transport
MES of Ukraine, c. Dnipropetrovsk
Ukraine, 49010, c. Dnipropetrovsk, Lazarian St., 2

Appliance of Genetic Algorithms in Solving Tasks of Data Structures Adaptation

В.І. Шинкаренко, Г.В. Забула
Дніпропетровський національний університет залізничного транспорту
МОН України, м. Дніпропетровськ
Україна, 49010, м. Дніпропетровськ, вул. акад. Лазаряна, 2

Застосування генетичного алгоритму у задачах адаптації структур даних

Представлена разработанная схема адаптации структур данных в оперативной памяти. Для поиска адаптированной структуры данных предложено применение генетического алгоритма. Определены его особенности. Для растровых изображений с соответствующими заголовками выполнены экспериментальные исследования временной эффективности адаптированных структур данных, времени сходимости генетического алгоритма и условия целесообразности применения адаптации.
Ключевые слова: генетический алгоритм, адаптация, структура данных, эксперимент.

The developed scheme of data structures adaptation in memory is given. To search for an adapted data structure, genetic algorithm is proposed and its characteristics are defined. For raster images with the appropriate headings, experimental studies of the time efficiency of adapted data structures, the convergence time of the genetic algorithm and conditions for feasibility of adapting are performed.
Key Words: genetic algorithm, adaptation, data structure, experiment.

Представлена розроблена схема адаптації структур даних в оперативній пам'яті. Для пошуку адаптованої структури даних запропоновано застосування генетичного алгоритму. Визначені його особливості. Для растрових зображень з відповідними заголовками виконані експериментальні дослідження часової ефективності адаптованих структур даних, часу збіжності генетичного алгоритму та умови доцільності застосування адаптації.
Ключові слова: генетичний алгоритм, адаптація, структура даних, експеримент.

Введение

Адаптация растений и животных к окружающей среде, человека к природным условиям жизни и социальной среде, как правило, основана на неинтеллектуальных механизмах. В искусственных программных средах адаптация невозможна без принятия соответствующих решений, т.е. без элементов искусственного интеллекта.

Современные программные, информационные системы все более интеллектуализируются, приобретая адаптивные способности. Адаптация широко применяется в интерфейсах пользователя, значительно реже в части обработки и преобразования данных, но практически не применяется при хранении данных на различных носителях. В данной работе, восполняя этот пробел, в развитие предложенного ранее подхода [1], решается задача структурной адаптации структур данных (СД) в оперативной памяти (ОП).

Временная эффективность обработки СД существенно зависит от способа представления данных в ОП, значения этих данных, порядка их обработки и программно-аппаратной среды эксплуатации. Это подтверждается особенностями обработки СД в оперативной памяти [2] и выполненными ранее компьютерными экспериментами [3].

Применение генетического алгоритма (ГА) для поиска наиболее подходящих СД обусловлено сложной зависимостью их временной эффективности от программно-аппаратной среды эксплуатации и порядка использования СД. Свойства этой зависимости неизвестны и сложно формализуемы. Как известно, ГА в таких случаях является наиболее подходящим инструментом [4], [5].

Цель данной работы – исследование эффективности применения предложенного ранее метода адаптации структур данных [1] и генетического алгоритма как его составной части.

Постановка задачи

Экспериментальными методами необходимо установить:

– насколько большим может быть преимущество адаптированной СД по отношению к традиционным и единообразным способам их представления;

– время сходимости ГА;

– условия, при которых адаптация будет в принципе полезной.

Преимущества адаптированной СД можно оценить с помощью показателей степени и области превосходства [1].

Схема адаптации

Порядок и составляющие процесса адаптации на основе IPO диаграмм [6] представлены в табл. 1 с пояснениями, приведенными ниже.

Таблица 1 – Процессы адаптации структур данных

Вход	Процесс	Выход
Диаграмма логической СД, формируемая в интерактивном режиме	Разработки MERA-диаграммы [1] средствами редактора диаграмм Dia [7]	Файл формата *.dia, содержащий логическую СД
Конкретный экземпляр данных, на которых выполняется адаптация; *.dia файл	Формирование XML файлов	*.xml файл
В составе генетического алгоритма		
.dia файл; шаблоны	Формирование хромосомы	Хромосома
Хромосома; поколение хромосом	Формирование поколений хромосом, отсеивание и отбор	Поколение хромосом; лучшая хромосома
*.dia файл; хромосома; набор шаблонов	Генерация физических реализаций	Конкретная физическая СД, соответствующая хромосоме
*.xml файлы; сценарий работы с данными; физическая реализация СД	Регистрация времени	Время выполнения физической реализации

Концептуальная модель логической структуры данных должна отображать составляющие, их свойства и связи между составляющими.

На рис. 1 показан пример такой модели. Данное представление основано на известных средствах ER-диаграмм [8] и представляют собой модифицированные ER-диаграммы, нагруженные атрибутами, – MERA-диаграммы. Особенности формирования логической структуры данных приведены в [1].

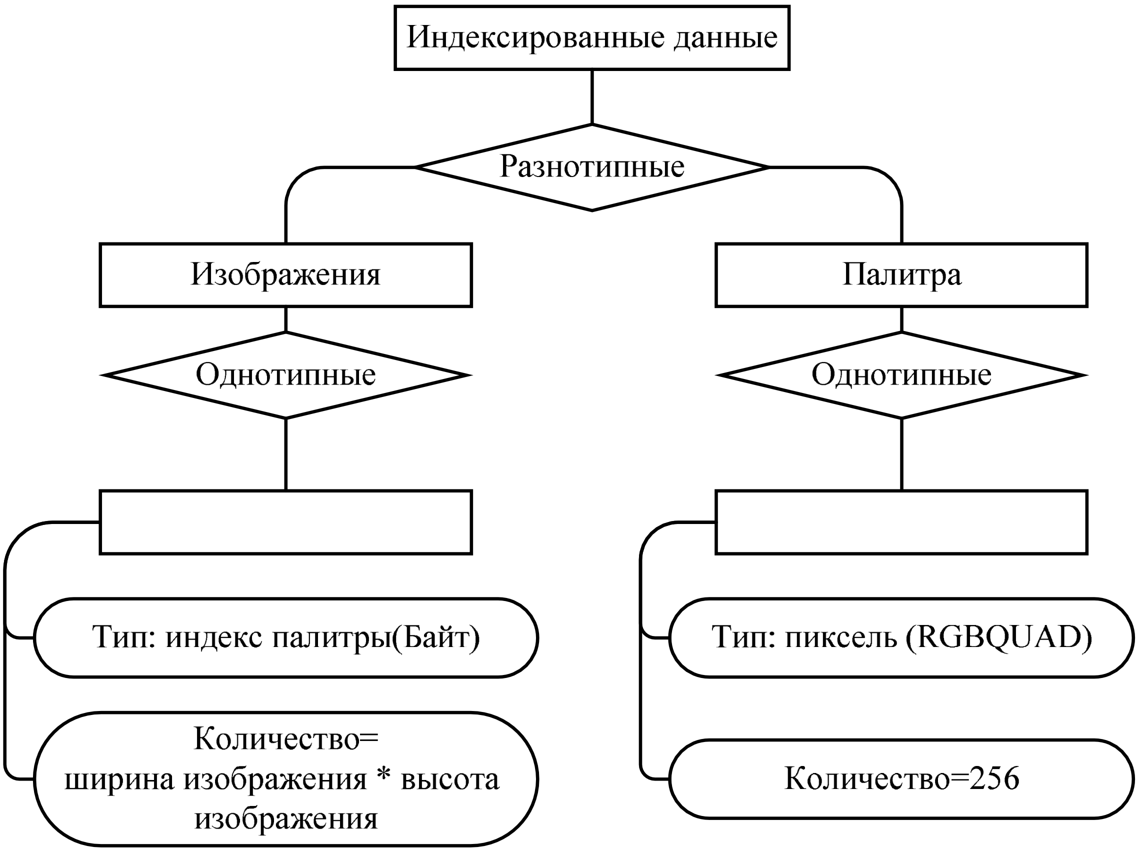


Рисунок 1 – Фрагмент логической структуры данных *.bmp файла

Структура данных на физическом носителе информации (в данном случае в оперативной памяти), т.е. порядок расположения элементов и методы доступа к ним, в соответствии с выбранным порядком определяют физическую структуру данных.

Для формирования конкретной физической СД применяем программные шаблоны [1]. Программный шаблон – полный набор методов доступа к элементам СД, которые определяют порядок размещения данных, обеспечивают формирование СД, добавление/удаление элементов, поиск и т.п. В данной работе реализованы такие шаблоны: динамический массив, связный список, связный список массивов, хэш-таблица.

Реализация конкретных физических СД выполняется на основе логической СД и шаблонов. Для каждой связи в логической СД выбирается один из шаблонов. В соответствии с выбранными шаблонами, средствами рефлексии [9] формируется программа обработки данных.

Порядок обработки данных, т.е. последовательность операций добавления, удаления, поиска и др., назван сценарием.

Сценарий может быть определен средствами стохастических грамматик [10], отслеживанием порядка обработки данных в эксплуатируемой системе, предельными количествами каждой операции обработки данных (с возможностью задания приоритетов операций и ограничений на их последовательность). В данной работе применен последний способ.

Особенности генетического алгоритма

Для адаптации структур данных применялись известные подходы [11] на основе генетического алгоритма [4] с особенностями, обусловленными спецификой решаемой задачи.

Представим детализацию обобщенной схемы работы ГА [5].

Оптимизируемыми параметрами являются физические реализации СД каждого отношения логической СД.

Кодирование хромосом выполнено в виде вектора чисел. Каждое число (аллель) представляет собой порядковый номер в списке шаблонов. Позиция числа (локус) в хромосоме обозначает индекс отношения, к которому применяется шаблон. Гены в хромосоме не являются зависимыми или взаимозаменяемыми. В старших битах чисел могут кодироваться дополнительные свойства шаблонов. Так, для связных массивов указывается количество элементов в каждом массиве.

Имея в логической структуре пять отношений и четыре шаблона, общее количество возможных физических реализаций – 1024 (4^5).

Целевой функцией является время выполнения набора сценариев на совокупности файлов, к которым выполняется адаптация.

Инициализация начальной популяции выполняется следующим образом. Каждая хромосома заполняется одинаковыми аллелями. Количество хромосом в первом поколении соответствует количеству различных шаблонов. Например, для пяти отношений и четырех шаблонов в первом поколении будет присутствовать четыре особи: [0, 0, 0, 0, 0], [1, 1, 1, 1, 1], [2, 2, 2, 2, 2], [3(25), 3(25), 3(25), 3(25), 3(25)]. Здесь 0 – динамический массив; 1 – связный список; 2 – хэш-таблица; 3 – связный список массивов, 25 – количество элементов в связных массивах. Для повышения разнообразия хромосом в первое поколение также добавляются несколько потомков существующих хромосом.

Оператор отбора выбирает хромосомы для следующего поколения и скрещивания со значением целевой функции выше среднего для хромосом текущего поколения.

Оператор скрещивания выполняется на основе нескольких родительских хромосом, количество которых задается параметром ГА.

Скрещивание генов хромосомы производится следующим образом: для каждого локуса генерируется случайное число, значение которого определяет номер родителя в наборе родительских хромосом:

$$y_{1,i} = x_{\xi_i,i},$$

где $X_i = [x_{i,1},\, x_{i,2} \ldots x_{i,n}]$ – родительские хромосомы, $Y_1 = [y_{1,1},\, y_{1,2} \ldots y_{1,n}]$ – дочерняя хромосома; ξ – вектор случайных чисел; n – кол-во отношений.

Пример скрещивания приведен на рис. 2.

Оператор мутации изменяет выборочные гены сложением по модулю m (кол-во шаблонов) текущего значения с единицей.

Критерий остановки – отсутствие изменений лучшей хромосомы в течение заданного количества поколений.

Параметры ГА алгоритма: вероятность участия хромосом в скрещивании, предельное количество поколений, минимальное количество хромосом в популяции, вероятность мутации, количество родительских хромосом для скрещивания, количество уникальных хромосом для скрещивания.

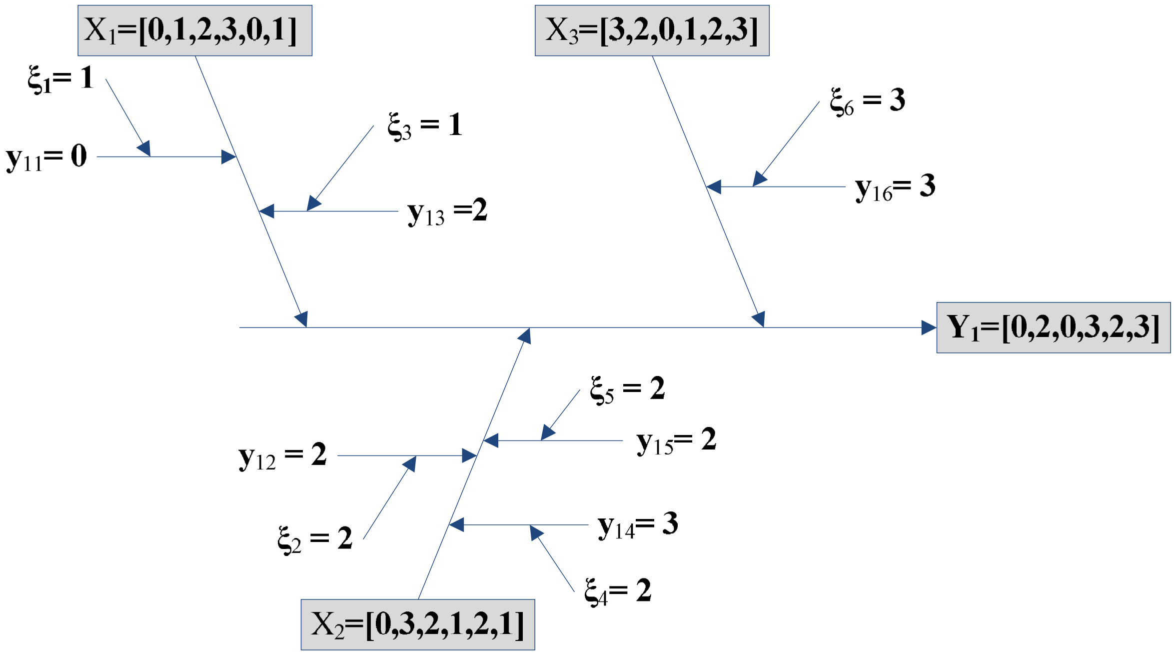


Рисунок 2 – Скрещивание хромосом X_1, X_2, X_3

Экспериментальные исследования адаптации

Для определения возможностей и оценки эффективности адаптации выполнен компьютерный эксперимент.

Цель эксперимента – оценка эффективности ГА и адаптации для растровых изображений, считанных из BMP файлов, в оперативной памяти.

Экспериментальная база состояла из пяти компьютеров, характеристики которых приведены в табл. 2, и исследуемой выборки из 33 файлов формата *.bmp размером 2…3 Мб. Объем данных подобран таким образом, чтобы он значительно превышал размер кэша данных. В противном случае любая физическая реализация будет обеспечивать максимальную временную эффективность структур данных.

Таблица 2 – Технические характеристики компьютеров

ПЭВМ	Процессор	Кэш L1 кода / L1 данных / L2, Кб	Тактовая частота / частота системной шины / частота памяти, МГц	Время доступа к ОП: чтение / запись, Мб/с	Операционная система
C_1	Intel Core 2 Duo E4600	32/32/ 2048	2 400 (12 × 200) / 800 / 400	5350 / 1962	Windows 7 Ultimate Prof
C_2	Intel Celeron G530 Dual Core	2×32/2×32/ 2×256	2 400 (24 × 100) / 100 / 532	7470 / 3590	Windows 7 Ultimate Prof
C_3	Intel Celeron E3400 3800 Dual Core	2×32/2×32/ 1024	2 600 (13 × 200) / 200 / 400	4250 / 1680	Windows 7 Ultimate Prof
C_4	Intel Mobile Core 2 Duo T5500	2×32/2×32/ 2048	1 666 (10 × 166) / 666 / 333	5850 / 1720	Windows 7 Ultimate
C_5	Intel Core 2 Dup E6500	2×32/2×32/ 2048	1 200 (6 × 200) / 800 / 333	4370 / 1200	Windows XP Prof

Логическая структура данных BMP файла [1] имеет шесть отношений, связывающих основные подструктуры: заголовок BMP файла (P_1), информационный заголовок (P_2), палитра (P_3), цвета палитры (P_4), данные изображения (P_5) и их связующая (P_6).

Формирование сценариев обработки данных выполнялось стохастическим методом с заданными границами для количества применяемых операций обработки данных, которые приведены в табл. 3.

Таблица 3 – Ограничения методов обработки данных

Подструктура	Метод	Количество включений метода в сценарии	
		Min	Max
P_1	Все 10 методов	0	0
P_2	Чтение ширины изображения в пикселях Остальные 21 метод	0 0	100 0
P_3	Все 8 методов	0	0
P_4	Добавление цвета в палитру Остальные 7 методов	0 0	256 0
P_5	Добавление пикселя к изображению Удаление пикселя из изображения Остальные 6 методов	10 000 1 000 0	20 000 2 000 0
P_6	Все 8 методов	0	0

В эксперименте выполнены исследования с применением трех сценариев: с ограничением, как в табл. 3, (A_1) и уменьшенными (A_0) и увеличенными (A_2) в 10 раз.

Всего выполнено 45 опытов (адаптаций) – на пяти компьютерах по сценариям с тремя наборами ограничений, повторяя каждый опыт трижды.

Выполнена оценка качества адаптированных структур данных. Для оценки того, насколько найденные адаптивные СД лучше однотипных, в том числе и общепринятых, определены *S*- и *R*-показатели временной эффективности [3] сценариев работы с данными, построенных по заданным ограничениям (табл. 3):

$$S_{ij} = [55{,}0^{+27{,}0}_{-27{,}0},\ 54{,}6^{+27{,}7}_{-27{,}7},\ 25{,}8^{+29{,}0}_{-0{,}0},\ 69{,}1^{+20{,}4}_{-20{,}4},\ 57{,}1^{+29{,}3}_{-29{,}3}],\ R_{ij} = 100^{+0}_{-0}.$$

Здесь S_{ij} – показатель степени превосходства [1] адаптированной структуры над контрольными, которая основана на временных значениях выполнения алгоритмов (сценариев) обработки структур данных; *i* – адаптированная структура, *j* = 1…4 – структуры, соответствующие хромосомам [0, 0, 0, 0, 0], [1, 1, 1, 1, 1], [2, 2, 2, 2, 2], [3(25), 3(25), 3(25), 3(25), 3(25)], *j* = 5 – случайным образом сформированная структура. Отметим, что показатель S_{ij} = 50% соответствует двукратному превосходству по времени одного сценария над другим.

Большие доверительные интервалы отражают существенный разброс разницы по времени выполнения сценариев обработки адаптивной и контрольных структур данных.

Как показал эксперимент, адаптированные структуры данных в параллельных опытах часто не совпадают, то есть получена достаточно низкая воспроизводимость. В табл. 4 показаны структуры данных (закодированные хромосомы), к которым наиболее часто сходился генетический алгоритм.

Разница между средним временем выполнения сценария на адаптированной структуре данных ($\bar{t}_a$) и контрольной структуре, сформированной на основе динамических массивов - хромосома [0, 0, 0, 0, 0] ($\bar{t}_0$), достаточно большая. Если в процессе эксплуатации предполагается, что количество подобных сценариев $N = \bar{T}_a / (\bar{t}_0 - \bar{t}_a)$ будет больше 5…150, то время, затраченное на адаптацию, будет компенсировано экономией времени работы алгоритма.

Таблица 4 – Оценка эффективности адаптации

ПЭВМ	Преобладающие адаптированные структуры (хромосома)	Сценарий	Среднее время адаптации $\overline{T}_a$, мин.	Средняя экономия времени на одном сценарии $\bar{t}_0 - \bar{t}_a$, с	Количество сценариев, при котором адаптация оправдана, *N*
C_1	[2,3(50),2,2,2,2] [2,3(25),2,2,2] [2,2,2,2,2]	A_0	17,6	20,7	51
		A_1	80,2	285,6	17
		A_2	556,6	5020	7
C_2	[2,0,2,2,2,2] [1,0,1,1,1,1] [0,3(100),0,0,0]	A_0	14,2	5,8	148
		A_1	46,9	54,0	53
		A_2	429,9	2614	10
C_3	[2,2,2,2,2] [2,0,2,2,2,2] [2,3(50),2,2,2,2]	A_0	17,5	21,5	49
		A_1	57,6	167,1	21
		A_2	518,6	5088	7
C_4	[2,0,2,2,2] [2,3(50),2,2,2,2]	A_0	22,9	29,3	47
		A_1	53,7	291,4	12
		A_2	938,0	6340	9
C_5	[2,2,2,2,2]	A_0	18,6	17,3	65
		A_1	73,8	228,2	20
		A_2	476,6	6163	5

Приведенные ранее результаты говорят о достаточно хорошей скорости сходимости ГА и качестве найденных структур данных. Лучшая хромосома появлялась в 3…5 поколении и ГА сходился за 7…25 поколений. Однако подобранные параметры ГА не являются оптимальными. Было бы целесообразным решение задачи нахождения оптимальных параметров ГА средствами другого ГА и на более представительной выборке.

Характеристика представленного метода адаптации

Рассмотрим характеристики адаптации структур данных по схеме, предложенной в [12].

Требования к качеству. Применяется для адаптации программ с преобладающей обработкой структур данных с повышенными требованиями к временным характеристикам.

Элементы внешней среды, к которым выполняется адаптация: потоки входных данных (наборы входных данных при многократном выполнении программ) и программно-аппаратная среда эксплуатации.

Адаптирующий алгоритм представлен в виде отдельного алгоритма на основе генетического.

Адаптация структур данных выполняется в процессе эксплуатации в автоматическом режиме.

Для оценки качества применен измерительный алгоритм, который является частью адаптирующего и измеряет время обработки данных в определенном порядке размещенных в оперативной памяти согласно сгенерированной структуре данных.

Метод адаптации можно квалифицировать как метод структурной адаптации одновременно алгоритма и структуры данных.

Приведенные эксперименты показали, что в разных средах эксплуатации повышается или, по крайней мере, не ухудшается временная эффективность структур данных.

Таким образом, представленный метод адаптации имеет все признаки полноценного адаптивного алгоритма.

Выводы

Разработанная методика адаптации структур данных [1] апробирована экспериментальными исследованиями адаптации СД для растровых изображений, считанных из BMP файлов, в оперативной памяти.

Установлено, что время выполнения стохастических сценариев обработки данных, адаптированных СД, в 1,5...3 раза меньше времени выполнения соответствующих сценариев на контрольных, в том числе общепринятых, СД.

В работе определено одно из ключевых условий, при которых адаптация является целесообразной. В зависимости от количества методов обработки данных в сценарии и конкретных программно-аппаратных сред эксплуатации СД адаптация оправдана, если количество подобных сценариев при эксплуатации превышает 5…150. Таким образом, предполагается достаточно быстрая окупаемость адаптации по временным затратам.

Для повышения эффективности ГА предполагается автоматизация подбора его параметров.

Литература

1. Шинкаренко В.И. Повышение временной эффективности структур данных в оперативной памяти на основе адаптации / В.И. Шинкаренко, Г.В. Забула // Проблеми програмування. – 2012. – № 2-3. – С. 211-218.
2. Касперски К. Техника оптимизации программ. Эффективное использование памяти / К. Касперски. – СПб. : БХВ-Петербург, 2003. – 464 с.
3. Шинкаренко В.И. Экспериментальные исследования алгоритмов в программно-аппаратных средах : монография / Шинкаренко В.И. – Д. : Изд-во Днепропетр. нац. ун-та ж.-д. трансп. им. акад. В. Лазаряна, 2009. – 279 с.
4. Гладков Л.А. Генетические алгоритмы / Гладков Л.А., Курейчик В.В., Курейчик В.М. – М. : Физматлит, 2006. – 320 с.
5. Субботин С.О. Неітеративні, еволюційні та мультиагентні методи синтезу нечітко логічних і нейромережевих моделей : монографія / С.О. Субботін, А.О. Олійник, О.О. Олійник. – Запоріжжя : ЗНТУ, 2009. – 375 с.
6. Зиглер К. Методы проектирования программных систем / Зиглер К. – М. : Мир, 1985. – 328 с.
7. Dia – GNOME Live! [Электронный ресурс]. – Режим доступа : http://live.gnome.org/Dia.
8. Chan P. The Entity-Relationship Model – Toward a Unified View of Data / P. Chan // ACM Transactions on Database Systems (TODS) : сб. –Нью-Йорк : ACM, 1976. –Т. 1. – С. 9-36.
9. Шилдт Г. C# 4.0: полное руководство / Шилдт Г. – М. : Вильямс, 2010. – 1056 с.
10. Ту Дж. Принципы распознавания образов / Дж. Ту, Р. Гонсалес. – М. : Мир, 1978. – 411 с.
11. Курейчик В.М. Поисковая адаптация: теория и практика / Курейчик В.М., Лебедев Б.К., Лебедев О.К. – Москва : Физматлит, 2006. – 272 с.
12. Шинкаренко В.И. Потенциальные возможности адаптации алгоритмов / В.И. Шинкаренко, Е.Г. Васецкий, М.М. Пятковский // Искусственный интеллект. – 2011. – № 4. – С. 232-242.

Literatura

1. Shinkarenko V. I. Problemy programuvannja. 2012. № 2-3. S. 211-218.
2. Kasperski K. Tehnika optimizacii program. Efektivnoje ispol’zovanije pamjati. 2003. 464 s.

3. Shinkarenko V. I. Eksperimental'noje issledovanije algoritmov v programmno-apparatnyh sredah. 2009. 279 s.
4. Gladkov L.A. Geneticheskije algoritmy. 2006. 320 s.
5. Subbotin S.O. Neiteratyvni, evoljucijni ta multiagentni metody sintezu nechitko logichnyx i nejromeregevyx modelej. 2009. 375 s.
6. Zigler K. Metody projektirovanija programmnyx system. 1985. 328 s.
7. Dia – GNOME Live!
8. Chan P. ACM Transactions on Database Systems. 1976. T.1. S.9-36.
9. Shildt G. C# 4.0: polnoje rukovodstvo. 2010. 1056 s.
10. Tu Dg. Principy raspoznavanija obrazov. 1978. 411 s.
11. Kurejchik V.M. Poiskovaja adaptacija: tjeorija I praktika. 2006. 272 s.
12. Shinkarenko V. I. Iskusstvennyj intellect. 2011. №.4. S. 232-242.

RESUME

V.I. Shynkarenko, H.V. Zabula

Using Genetic Algorithms in Solving Tasks of a Data Structures Adaptation

The paper presents the scheme designed to adapt data structures in memory. Genetic algorithm is proposed for search an adapted data structure.

Computer experiment was performed for evaluating genetic algorithm and adaptation of raster images read from bmp files efficiency.

Experiment base consists of five PCs and selection of the 33 bmp files from two to three Mb size. Performed 45 tests (adaptations) are on five PCs using three scenarios with three different limitations repeating each thrice.

For data structure adaptation, the known methods of using genetic algorithm with specific changes caused by problem were applied.

It is determined that executing stochastic scenarios for data processing using adapted data structure in 1.5…3 times is less than using common data structures.

The paper defines one of key conditions when adaptation is advisable. Depending on data processing method count in scenario, specific hardware and software environment where data structures is using adaptation is advisable when scenario number is over 5…150. In this case it is assumed fast time payback for adaptation time consumption.

Статья поступила в редакцию 06.06.2012.